Cross Parity Check Convolutional Codes

TOM FUJA, MEMBER, IEEE, CHRIS HEEGARD, MEMBER, IEEE, AND MARIO BLAUM, MEMBER, IEEE

Abstract—A class of convolutional codes called cross parity check (CPC) codes, useful for the protection of data stored on magnetic tape, is described and analyzed. CPC codes are first explained geometrically; their construction is described in terms of constraining data written onto a tape in such a way that, when lines of varying slope are drawn across the tape, the bits falling on those lines sum to zero modulo two. This geometric interpretation is then formalized by the construction of canonical parity check matrices and systematic generator matrices for CPC codes and by computing their constraint lengths. The distance properties of CPC codes are analyzed, and it is shown that these codes are maximum distance separable (MDS) convolutional codes. In addition, examples are given of both error and erasure decoding algorithms that take advantage of the geometric regularity of cross parity check codes. Finally, the technique of parity check matrix reduction—useful for reducing the inherent decoding delay of CPC codes—is described. This technique consists of dividing each term of the parity check matrix by some polynomial and retaining only the remainder. A class of polynomials which are particularly attractive for this purpose is identified.

I. INTRODUCTION

THIS PAPER describes a class of convolutional codes called cross parity check (CPC) codes. The intended application of these codes is the protection of data stored on magnetic tape.

Until recently, most error control schemes implemented on magnetic tape involved the use of block codes—most typically, a Reed–Solomon (RS) or other BCH code [1]–[3]. However, the recent literature [4]–[7] has provided examples of how convolutional codes can be used in this capacity; the error control techniques described in [4]–[7] are based on simple geometric ideas—specifically, the imposition of parity constraints on data falling on lines of varying slope throughout the tape. One attractive advantage such techniques have over block codes is that they avoid the expensive necessity of performing operations over nonbinary fields; all syndrome computations can be implemented with XOR gates.

Cross parity check codes evolved from the codes described in [4]–[7]. Our goal has been to examine the algebraic structure of such "geometrically inspired" convolutional codes to see what kinds of insights can be gained. In this paper we formalize the geometric interpretation of CPC codes and examine their distance properties. In addition, we give examples of ways in which the geometric regularity of cross parity check codes can be used to provide simple decoding algorithms. Finally, we introduce the idea of *parity check matrix reduction* as a means of constructing new convolutional codes from old ones.

A Summary of Results

In this paper we propose and analyze a class of "geometrically oriented" convolutional codes with one particular application in mind—the protection of data stored on magnetic tape. Toward this end, the following results are established.

1) A set of codes called cross parity check codes are defined. The particular code CP(n, k, m) is an (n, k) convolutional code with the following geometric interpretation. If the n binary sequences making up a codeword of CP(n, k, m) are written onto the n tracks of a magnetic tape, then the bits falling on every line of slope $1/j$ must sum to zero modulo two, where j takes on the $n-k$ different values, $j = m, m-1, m-2, \cdots, m-(n-k-1)$. This geometric interpretation is formalized by considering CP(n, k, m) as a k-dimensional subspace of n-tuples of a Laurent series with binary coefficients; a nicely "canonical" parity check matrix for CP(n, k, m)—that is, a matrix whose null space is CP(n, k, m) and whose rows reflect the geometric constraints—is identified.

2) For magnetic tape channels, the most useful measure of performance is not free distance but minimum distance. Cross parity check codes are optimal in the following sense: CP(n, k, m) has minimum distance $n-k+1$, the largest possible value of any (n, k) convolutional code; in other words, the Singleton bound is achieved, and so CPC codes are maximum distance separable.

3) The geometric regularity in cross parity check codes makes it possible to compute a closed-form systematic generator matrix for CP(n, k, m). In addition, the minimal constraint length of CP(n, k, m)—a measure of the complexity of the encoding operation—is computed as $k\binom{n-k}{2} - km(r-1-m)$.

4) A general erasure decoding algorithm for CP(n, k, m) is described, along with a two-error correcting algorithm for CP$(n, n-4, 1)$.

5) Parity check matrix reduction—dividing each term in a parity check matrix by some polynomial in D and

Manuscript received September 7, 1987; revised September 8, 1988. This work was supported in part by the National Science Foundation Engineering Research Center Program under Grant NSFD CDR 8803012 and in part by the National Science Foundation under Engineering Grant ECS83-52220. This paper was presented in part by the IBM Workshop on Coding Theory, San Jose, CA, August 11–12, 1986, and in part at the IEEE International Symposium on Information Theory, Ann Arbor, MI, October 6–9, 1986.

T. Fuja is with the Department of Electrical Engineering, Systems Research Center, University of Maryland, College Park, MD 20742.

C. Heegard is with the School of Electrical Engineering, Cornell University, Ithaca, NY 14853.

M. Blaum is with the IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099.

IEEE Log Number 8931521.

0018-9448/89/1100-1264$01.00

retaining only the remainder—is introduced as a means of constructing new convolutional codes from old ones; the motivation here is the reduction of the delay inherent in decoding CPC codes. It is shown that, if a CPC code is "reduced" with a polynomial of the form D^j+1 for odd $j \geq n$, then the resulting convolutional code is maximum distance separable *and* retains a certain geometric regularity.

II. Definitions and Background

We begin by giving a geometric interpretation of cross parity check codes; we then use that interpretation to define CPC codes formally in terms of a parity check matrix. In addition, we compare CPC codes with similarly defined error control codes described in the literature and outline the problems to be considered in this paper.

A. A Geometric Interpretation of Cross Parity Check Codes

An n-track binary magnetic tape can be modeled as a strip A_n, where

$$A_n = \{ a_{ij} : 0 \leq i \leq n-1,\ -\infty < j < \infty,\ a_{ij} \in \{0,1\} \}.$$

The interpretation is that for fixed i $(0 \leq i \leq n-1)$, $\{a_{ij} : -\infty < j < +\infty\}$ are the contents of the ith track on the tape. It is assumed that there exists some j_0 such that $a_{ij} = 0$ for all i and for all $j < j_0$; that is, the tape has some beginning, before which all the data are assumed to be zero.

Definition: Consider an n-track magnetic tape, as already described. We say such a tape is encoded with CP(n, k, m) if

$$\sum_{i=0}^{n-1} a_{i,\, j+\alpha i} = 0$$

for all j, $-\infty < j < +\infty$, and for $\alpha = m, m-1, \cdots, m-(n-k-1)$. (Addition is modulo-two.)

The geometric interpretation of these codes consists of constraining the data written onto the tape so that when lines of varying slope are drawn across the tape, the bits falling on those lines sum to zero modulo two. Specifically, for a CP(n, k, m) encoded tape, we require that the bits on a line of slope x across the tape sum to zero modulo two, where

$$x \in \left\{ \frac{1}{m}, \frac{1}{m-1}, \cdots, \frac{1}{m-(n-k-1)} \right\}.$$

(In our notion of "slope," we assume that track 0 is the "bottom" track and track $n-1$ is the "top"; in addition, a slope of $+\infty$ (i.e., $1/0$) corresponds to a parity check straight across the tape.) Thus there are $n-k$ slope constraints; m is the parameter that tells us how many of these constraints involve positive slopes. We will see later in this section that these $n-k$ constraints can be imposed through the addition of $n-k$ redundant tracks; thus a magnetic tape encoded with CP(n, k, m) contains k data tracks and $r = n-k$ redundant tracks, and so CP(n, k, m) has a *rate* of k/n. Fig. 1 shows the parity check patterns for three CPC codes.

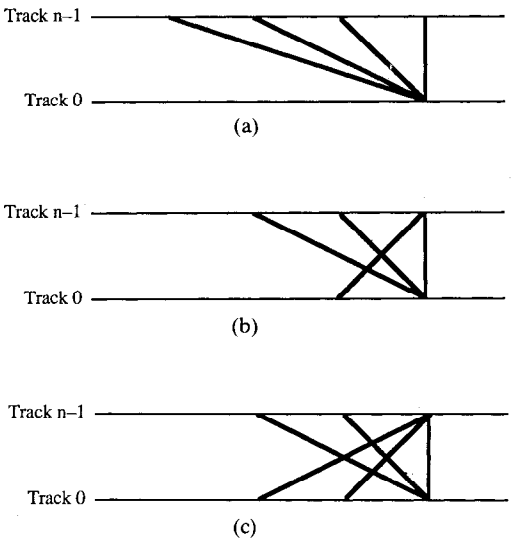

Fig. 1. Some examples of parity check patterns for CPC codes. (a) CP$(n, n-4, 0)$. (b) CP$(n, n-4, 1)$. (c) CP$(n, n-5, 2)$.

B. A Formal Description of Cross Parity Check Codes

Let $\boldsymbol{F} = \{0,1\}$ be the binary field, and define $\boldsymbol{F}[D]$ to be the set of all polynomials in D with coefficients in $\boldsymbol{F}$; further, let $\boldsymbol{F}(D)$ denote the field of quotients of $\boldsymbol{F}[D]$, consisting of all ratios of polynomials in $\boldsymbol{F}$ with nonzero denominators [10], [11]. In a similar vein, let $\boldsymbol{F}[[D]]$ be the set of power series over $\boldsymbol{F}$; that is,

$$\boldsymbol{F}[[D]] = \left\{ f(D) = \sum_{i=0}^{\infty} a_i D^i :\ a_i \in \{0,1\} \right\}.$$

We can extend $\boldsymbol{F}[[D]]$ to a field consisting of the quotients of $\boldsymbol{F}[[D]]$. This field is called the Laurent series over $\boldsymbol{F}$, and we denote it by $\boldsymbol{F}((D))$. It can be shown [18] that this field is isomorphic to the field consisting of all "one-sided" sequences of ones and zeros. That is,

$$\boldsymbol{F}((D)) = \left\{ f(D) = \sum_{i=r}^{\infty} a_i D^i :\ a_i \in \{0,1\},\ r \in Z \right\}.$$

Since convolutional codes are simply sets of "permissible" n-tuples of binary sequences, we can describe them equivalently as sets of n-tuples over $\boldsymbol{F}((D))$. We use this equivalence throughout this paper; when we speak of "writing" a Laurent series $f(D)$ onto a magnetic tape track, we mean storing the associated binary sequence as the track contents.

The last structure of interest will be $\boldsymbol{F}_{rz}(D)$, the set of realizable transfer functions. $\boldsymbol{F}_{rz}(D)$ is a subring of $\boldsymbol{F}(D)$ and consists of those ratios of polynomials whose denominators (after reduction to lowest terms) are not divisible by D; equivalently, $\boldsymbol{F}_{rz}(D)$ is made up of those functions that are in both $\boldsymbol{F}(D)$ and $\boldsymbol{F}[[D]]$—i.e., $\boldsymbol{F}_{rz}(D) = \boldsymbol{F}(D) \cap \boldsymbol{F}[[D]]$. It can be shown [10] that these are exactly the transfer functions that can be realized by causal finite state circuits; that is, $g(D) \in \boldsymbol{F}_{rz}(D)$ if and only if it is possible to construct a circuit such that if $x(D) \in \boldsymbol{F}((D))$ is the input, then $x(D)g(D)$ is the output.

Definition: An (n, k) binary convolutional code is any k-dimensional subspace of $\boldsymbol{F}^n((D))$ over $\boldsymbol{F}((D))$ that has a basis in $\boldsymbol{F}_{rz}^n(D)$.

Note that this definition is equivalent to the "usual" definition of an (n, k) binary convolutional code as the set of all possible outputs of a k-input n-output time-invariant linear causal finite-state sequential binary circuit [10].

Definition: For any (n, k) convolutional code $\boldsymbol{C}$, let $\boldsymbol{H}$ be a (nonunique) $(n-k)\times n$ matrix over $\boldsymbol{F}_{rz}(D)$ such that

$$\boldsymbol{C} = \left\{ \bar{c} = [c_0, c_1, \cdots, c_{n-1}] \in \boldsymbol{F}^n((D)) \colon \bar{c}\boldsymbol{H}^T = 0 \right\}.$$

Then $\boldsymbol{H}$ is a *parity check matrix* for $\boldsymbol{C}$.

A parity check matrix for CP(n, k, m) can be constructed as follows; let

$$\boldsymbol{H}_0 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & D & D^2 & \cdots & D^{n-1} \\ 1 & D^2 & D^4 & \cdots & D^{2(n-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & D^{n-k-1} & D^{2(n-k-1)} & \cdots & D^{(n-k-1)(n-1)} \end{bmatrix}.$$

This matrix is a parity check matrix for CP$(n, k, m = 0)$. To see this, suppose that $\bar{c} = [c_0, c_1, \cdots, c_{n-1}] \in \boldsymbol{F}^n((D))$ satisfies $\bar{c}\boldsymbol{H}^T = \boldsymbol{0}$, and further that c_i is given by

$$c_i = \sum_{j=r_i}^{\infty} c_{i,j} D^j.$$

Then the first row of $\boldsymbol{H}_0$ implies that $\sum_{i=0}^{n-1} c_{i,j} = 0$ for all j. That is, if c_i is written onto the ith track of a magnetic tape for $i = 0, 1, \cdots, n-1$, then the bits falling on any line drawn straight across the tape sum to zero modulo two; the "infinite" slope constraint is met. Similarly, the second row of $\boldsymbol{H}_0$ imposes the slope -1 constraint, the third row imposes the $-1/2$ constraint, and so on until the bottom row of $\boldsymbol{H}_0$ imposes the constraint that the bits falling on any line of slope $-1/(n-k-1)$ must sum to zero modulo two.

In a similar way we can construct "canonical" parity check matrices that reflect the geometric constraints for CP(n, k, m) for nonzero m; we will denote this matrix by $\boldsymbol{H}_m$. As examples, consider the three CPC codes shown in Fig. 1; the canonical parity check matrix for CP$(n, n-4, 0)$ is given by

$$\boldsymbol{H}_0 = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & D & D^2 & \cdots & D^{n-1} \\ 1 & D^2 & D^4 & \cdots & D^{2(n-1)} \\ 1 & D^3 & D^6 & \cdots & D^{3(n-1)} \end{bmatrix},$$

the canocial parity check matrix for CP$(n, n-4, 1)$ is given by

$$\boldsymbol{H}_1 = \begin{bmatrix} D^{n-1} & D^{n-2} & D^{n-3} & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & D & D^2 & \cdots & D^{n-1} \\ 1 & D^2 & D^4 & \cdots & D^{2(n-1)} \end{bmatrix},$$

and the one for CP$(n, n-5, 2)$ is

$$\boldsymbol{H}_2 = \begin{bmatrix} D^{2(n-1)} & D^{2(n-2)} & D^{2(n-3)} & \cdots & D^2 & 1 \\ D^{n-1} & D^{n-2} & D^{n-3} & \cdots & D & 1 \\ 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & D & D^2 & \cdots & D^{n-1} & D^{n-1} \\ 1 & D^2 & D^4 & \cdots & D^{2(n-1)} & D^{2(n-1)} \end{bmatrix}.$$

In general, we can obtain $\boldsymbol{H}_m$ from $\boldsymbol{H}_0$ as follows. Let $\boldsymbol{\Psi}_n$ be the $n \times n$ matrix

$$\boldsymbol{\Psi}_n = \operatorname{diag}[1, D, D^2, D^3, \cdots, D^{n-1}]$$

and for m, $0 \le m \le n-k-1$, define the $(n-k)\times(n-k)$ matrix $\boldsymbol{\Phi}_m$ given by

$$\boldsymbol{\Phi}_m = \operatorname{diag}[D^{m(n-1)}, D^{(m-1)(n-1)}, D^{(m-2)(n-1)}, \cdots, D^{(n-1)}, 1, 1, \cdots, 1].$$

(Here, $\operatorname{diag}[x_1, x_2, \cdots, x_j]$ is defined as the $j \times j$ diagonal matrix with $\{x_1, \cdots, x_j\}$ on the diagonal.) Then, the canonical parity check matrix for CP(n, k, m) $(0 \le m \le n-k-1)$ is given by

$$\boldsymbol{H}_m = \boldsymbol{\Phi}_m \boldsymbol{H}_0 \boldsymbol{\Psi}_n^{-m}.$$

(*Note:* The constraint that m be at least zero, and no greater than $n-k-1$ is, in our geometric description, equivalent to requiring that one of the parity check lines be straight across the track (i.e., have a slope of $+\infty$). While the generalization to larger or smaller m is obvious, it is of questionable practical value, and so we henceforth assume that $0 \le m \le n-k-1$.)

$\boldsymbol{H}_m$ is of full rank; that is, its rows span a subspace of dimension $n-k$. (In fact, in Section III-B we will prove the stronger fact that every $n-k$ columns of $\boldsymbol{H}_m$ are linearly independent.) This means that CP(n, k, m)—defined as the null space of $\boldsymbol{H}_m^T$—is of dimension k, and so CP(n, k, m) is an (n, k) binary convolutional code. This verifies our earlier claim that by adding $r = n-k$ redundant tracks to k data tracks, the $n-k$ slope constraints of CP(n, k, m) can be imposed.

C. Previous Work

Cross parity check codes were motivated by and evolved from error control techniques devised by Prusinkiewicz and Budkowski [4], Patel [5], [6], and Blaum [7]. One goal of our research has been to take "geometrically inspired" error control techniques such as those described in [4]–[7] and place them firmly within the context of convolutional codes by considering their algebraic structure.

Like CPC codes, the techniques described in these earlier papers were defined in terms of constraining data so as to meet parity slope conditions. Prusinkiewicz and Budkowski [4] described a code for magnetic tape that is essentially a "blocked" version of CP$(n, n-2, 0)$. Patel developed an error control technique that is geometrically similar to CP$(n, n-3, 1)$ [5]; he then altered that technique to construct a code that was implemented on IBM's 3480 tape subsystem [6]. Blaum [7] used similar ideas to describe a class of convolutional codes that bear some resemblance to the class of "balanced" CPC codes—i.e., CP$(n, k, \lfloor (n-k)/2 \rfloor)$.

One difference between CPC codes and the codes described in [5]–[7] is that the parity check lines defining the codes in [5]–[7] are not taken over the entire width of the tape. This was done to provide a more transparent encoding process; however, it came at the expense of a more complex decoder.

Finally, note that, concurrently with the evolution of the codes described earlier, papers appeared concerning so-called *orchard codes* [8], [9]. These form a class of rate $(n-1)/n$ codes that bear some resemblance to CPC codes in that they are described in terms of meeting a single geometric parity constraint.

D. Problems Addressed

In this paper we consider the following problems.

1) What distance properties are important for codes to be used in longitudinal magnetic tape applications? How do CPC codes compare to other convolutional codes with respect to these properties?
2) How can we construct encoders for CPC codes? This problem is important because, as mentioned in Section II-C, earlier "geometric" convolutional codes [5]–[7] were designed to make the encoding process trivial; this, however, resulted in a nonoptimal design as far as decoding and distance properties. Since CPC codes were designed without regard for the encoder, the effect of the design on the encoding process should be investigated.
3) How complex are CPC encoders? What are their constraint lengths?
4) How can the geometric properties of CPC codes be used for decoding?
5) How can the method of parity check matrix reduction—introduced by Piret and Krol [13]—be used to construct other MDS convolutional codes with geometric regularity?

III. Important Distance Properties of Parity Check Codes

In Section II the geometric nature of cross parity check codes was formalized by the introduction of certain algebraic structures and the construction of canonical parity check matrices. In this section this formality will be used to investigate the distance properties of CPC codes.

A. On the Minimum Distance of a Convolutional Code

On a longitudinal magnetic tape system, data are stored at a very high linear density on tracks that are relatively far apart from one another. This arrangement means that errors tend to come in bursts along tracks, with little correlation of errors between tracks. Piret and Krol [13] have suggested that, for codes to be used on channels exhibiting this kind of behavior, the important parameter is not free distance but rather what they call minimum distance.

Definition: The track weight of an n-tuple over any field is the number of nonzero components in that n-tuple.

(*Note:* We indicate the track weight of the n-tuple $\bar{u}$ by $|\bar{u}|$; from this definition it is obvious that $0 \le |\bar{u}| \le n$.)

Definition: The *track* distance between any two n-tuples is the track weight of their difference.

Definition: The *minimum* distance of a convolutional code is the smallest track distance between any two codewords in that code; for any code C denote the minimum distance of C by $d(C)$; then

$$d(C) = \min\{|\bar{x} - \bar{y}| : \bar{x}, \bar{y} \in C,\ \bar{x} \neq \bar{y}\}$$
$$= \min\{|\bar{u}| : \bar{u} \in C,\ \bar{u} \neq 0\}.$$

Note that this definition is different from the "usual" one for minimum distance; more commonly, the minimum distance of a convolutional code is defined as the Hamming weight of the lightest (in the Hamming weight sense) segment of any codeword, where the length of the segment is equal to the constraint length of the code. (See, for instance, [17].) The definition of minimum distance used here is the one given by Piret and Krol [13] and is analogous to the use of the same term for *block* codes.

Consider, then, an n-track magnetic tape encoded with an (n, k) convolutional code C of minimum distance $d(C)$. Assume that a possibly noisy version of this tape is read and that all the errors caused by the noise occur on $s \le n$ of the tracks; that is, s tracks contain errors and $n-s$ tracks are error-free. Then, as long as $s \le t = \lfloor (d(C)-1)/2 \rfloor$, there is exactly one codework in C that is a track distance of at most t from the received n-tuple. We will call such a code a t-error-correcting convolutional code because if all the errors are restricted to at most t tracks, then it is possible to recover the original codeword by simply mapping the received n-tuple to the closest (in the track-distance sense) codeword. In a similar way, if we designate a track as being *erased* when it has been labeled unreliable by an external source, it can be shown that a convolutional code C is capable of simultaneously correcting e erased tracks and t unerased tracks containing errors provided $2t + e \le d(C) - 1$.

B. Cross Parity Check Codes are Maximum Distance Separable

Consider an (n, k) convolutional code C with parity check matrix $\boldsymbol{H}$. The minimum distance $d(C)$ is equal to the smallest number of linearly dependent columns of $\boldsymbol{H}$. Since the number of linearly independent columns in a

matrix is equal to the number of linearly independent rows [12], and $\boldsymbol{H}$ has only $n-k$ rows, we have the following inequality, known as the *Singleton bound*:

$$d(C) \le n-k+1. \tag{1}$$

Any convolutional code for which (1) holds with equality is called a maximum distance separable (MDS) convolutional code [13].

Theorem 1: CP(n, k, m) is MDS.

Proof: Equation (1) holds with equality if and only if a parity check matrix for C is *totally non-singular*—that is, if an only if each $(n-k)\times(n-k)$ subarray of a parity check matrix is invertible. Thus we want to show that each of the $\binom{n}{n-k}$ different $(n-k)\times(n-k)$ minors of $\boldsymbol{H}_m$ is nonzero. (Here, an $r\times r$ *minor* of an $i\times j$ matrix is the determinant of an $r\times r$ subarray of that matrix.) To this end define for an arbitrary $p\times q$ matrix A the $p\times j$ matrix $A[i_0, i_1, \cdots, i_{j-1}]$ obtained by selecting columns $i_0, i_1, \cdots, i_{j-1}$ from A. (Here, $0 \le i_0 < i_1 < \cdots < i_{j-1} < q$.) Then any subarray of $\boldsymbol{H}_m$ can be factored into

$$\boldsymbol{H}_m[i_0, i_1, \cdots, i_{n-k-1}] = \boldsymbol{\Phi}_m \boldsymbol{H}_0[i_0, i_1, \cdots, i_{n-k-1}]\Lambda^m$$

where $\Lambda = \operatorname{diag}[D^{-i_0}, D^{-i_1}, \cdots, D^{-i_{n-k-1}}]$. Furthermore, $\boldsymbol{\Phi}_m$ and Λ^m are both $(n-k)\times(n-k)$ diagonal matrices with nonzero determinant; thus we can restrict our attention to matrices of the form $\boldsymbol{H}_0[i_0, i_1, \cdots, i_{n-k-1}]$. However,

$$\det\left(\boldsymbol{H}_0[i_0, \cdots, i_{n-k-1}]\right)$$

$$= \det\begin{bmatrix} 1 & 1 & \cdots & 1 \\ D^{i_0} & D^{i_1} & \cdots & D^{i_{n-k-1}} \\ D^{2i_0} & D^{2i_1} & \cdots & D^{2i_{n-k-1}} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ D^{(n-k-1)i_0} & D^{(n-k-1)i_1} & \cdots & D^{(n-k-1)i_{n-k-1}} \end{bmatrix}$$

$$= \prod_{a>b} \left(D^{i_a} + D^{i_b}\right) \neq 0$$

because the above determinant is Vandermonde. Thus all $(n-k)\times(n-k)$ minors of $\boldsymbol{H}_m$ are nonzero, and the theorem follows. Q.E.D.

It is worth noting that the convolutional code implemented on IBM's 3480 tape subsystem [6] is not MDS; even though it uses four parity tracks, it is only capable of correcting *half* of the two-error configurations. This is because the fourth parity constraint was formed by "breaking" the vertical slope constraint into two parity checks, each going halfway across the tape; this means that if two errors occur in the same half of the tape, then the configuration is not correctable.

IV. Encoding Cross Parity Check Codes

In this section we address two issues related to the construction of encoders for cross parity check codes. First, a class of systematic generator matrices for CPC codes is developed. Then, the minimal constraint length of CP(n, k, m)—a measure of the complexity of the encoding process—is calculated. In each case, the highly regular structure of cross parity check codes makes it possible to compute closed-form expressions for quantities which in general may only be computed iteratively.

A. Systematic Generator Matrices for CPC Codes

We begin by reviewing some terminology and ideas regarding convolutional code generators.

Definition: A generator for an (n, k) convolutional code is any $k\times n$ matrix over $\boldsymbol{F}_{rz}(D)$ such that the rows span the code over $\boldsymbol{F}((D))$.

We will concentrate our effort on the construction of a *systematic* generator. Such a generator is one which, when implemented, causes the k message sequences to be reproduced exactly in the code sequences. One way this can be accomplished is by embedding a $k\times k$ identity matrix in the generator matrix. To this end, for any given n, any $k<n$, any m, $0 \le m \le n-k-1$, and any x, $0 \le x \le r = n-k$, define $\boldsymbol{G}_{m,x}$ to be the $k\times n$ matrix over $\boldsymbol{F}(D)$ such that

a) CP$(n, k, m) = \{\bar{w}\boldsymbol{G}_{m,x}: \bar{w} \in \boldsymbol{F}^k((D))\}$,
b) $\boldsymbol{G}_{m,x} = [\bar{z}_0 \bar{z}_1 \cdots \bar{z}_{x-1} \boldsymbol{I}_k \bar{z}_x \cdots z_{r-1}]$

where $\boldsymbol{I}_k$ is the $k\times k$ identity matrix, and the $\bar{z}_j$'s are column k-vectors over $\boldsymbol{F}(D)$. $\boldsymbol{G}_{m,x}$ is a matrix with rows spanning CP(n, k, m) and containing the identity matrix in columns x through $x+k-1$. The goal is to choose x so that all the components of all the $\bar{z}_j$'s are realizable, and thus make $\boldsymbol{G}_{m,x}$ a valid systematic generator matrix for CP(n, k, m). This goal is realized in the following theorem.

Theorem 2: The matrix $\boldsymbol{G}_{m,x}$ as defined earlier is realizable for $x = r-m$.

Proof: The proof is given in the Appendix. Here we extract the portions of the proof that are necessary to give a closed-form expression for $\boldsymbol{G}_{m,r-m}$.

First, consider the case $m=0$. If we define $\boldsymbol{Z}$ to be the $k\times r$ matrix consisting of $\boldsymbol{G}_{0,x}$ with the identity removed, then it is shown in the Appendix that

$$\boldsymbol{Z} = \boldsymbol{Q}_{k,r}\boldsymbol{\Psi}_r^x \boldsymbol{P}^{-1}, \tag{2}$$

where $\boldsymbol{\Psi}_r = \operatorname{diag}[1, D, D^2, \cdots, D^{r-1}]$, $\boldsymbol{Q}_{a,b}$ is the $a\times b$ matrix

$$\boldsymbol{Q}_{a,b} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & D & D^2 & \cdots & D^{b-1} \\ 1 & D^2 & D^4 & \cdots & D^{2(b-1)} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & D^{a-1} & D^{2(a-1)} & \cdots & D^{(a-1)(b-1)} \end{bmatrix} \tag{3}$$

and

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{Q}_{x,r} \\ \hline \boldsymbol{Q}_{r-x,r}\boldsymbol{\Psi}_r^{x+k} \end{bmatrix}.$$

Most of the Appendix consists of a proof that $\boldsymbol{Z}$, as formulated in (2), is realizable.

For nonzero m, we claim that

$$\boldsymbol{G}_{m,x} = D^{-xm}\boldsymbol{\Psi}_k^{-m}\boldsymbol{G}_{0,x}\boldsymbol{\Psi}_n^{m}, \tag{4}$$

where $\boldsymbol{\Psi}_i = \mathrm{diag}[1, D, D^2, \cdots, D^{i-1}]$. This can be readily verified, since

$$\begin{aligned}\boldsymbol{G}_{m,x}\boldsymbol{H}_m^T &= D^{-xm}\boldsymbol{\Psi}_k^{-m}\boldsymbol{G}_{0,x}\boldsymbol{\Psi}_n^{m}\left[\boldsymbol{\Phi}_m\boldsymbol{H}_0\boldsymbol{\Psi}_n^{-m}\right]^T\\ &= D^{-xm}\boldsymbol{\Psi}_k^{-m}\boldsymbol{G}_{0,x}\boldsymbol{H}_0^T\boldsymbol{\Phi}_m\\ &= 0,\end{aligned}$$

and the multiplication leaves columns x through $x+k-1$ unchanged. It is shown in the Appendix that $\boldsymbol{G}_{m,x}$ is realizable for $x = r - m$. Q.E.D.

For the rest of this paper, we will define for a given n, $k < n$, and m, $0 \le m \le n-k-1$, the matrix $\boldsymbol{G}_m$ to be $\boldsymbol{G}_{m,r-m}$. Thus, by Theorem 2, $\boldsymbol{G}_m$ is a valid systematic generator for CP(n, k, m).

At this point it is illuminating to consider an example. We begin by constructing a generator matrix for CP(5,2,0). From Theorem 2, we know that to construct a systematic generator for such a code, we embed an identity matrix in columns 3 and 4; hence, from (2),

$$\boldsymbol{Z} = \begin{bmatrix}1 & 1 & 1\\ 1 & D & D^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & D^3 & 0\\ 0 & 0 & D^6\end{bmatrix}\begin{bmatrix}1 & 1 & 1\\ 1 & D & D^2\\ 1 & D^2 & D^4\end{bmatrix}^{-1}$$

$$= \begin{bmatrix}D^3 & D+D^2+D^3 & 1+D+D^2\\ D^3+D^4+D^5 & D+D^5 & 1+D+D^3+D^4\end{bmatrix}.$$

Thus our systematic generator for CP(5,2,0) is

$$\boldsymbol{G}_0 = \begin{bmatrix}D^3 & D+D^2+D^3 & 1+D+D^2 & 1 & 0\\ D^3+D^4+D^5 & D+D^5 & 1+D+D^3+D^4 & 0 & 1\end{bmatrix}.$$

A circuit that implements this encoder is shown in Fig. 2.

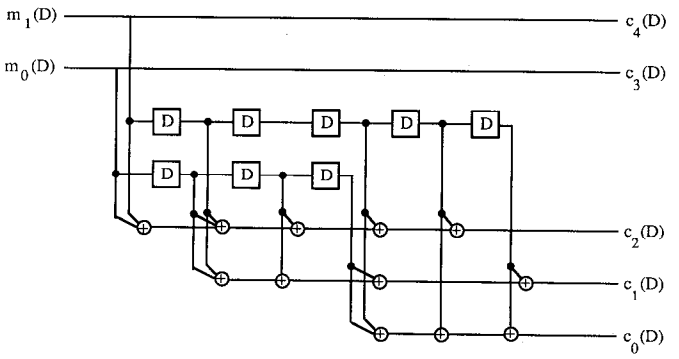

Fig. 2. Encoding circuit for CP(5,2,1).

Similarly, we can use the results from this section to construct a systematic generator for CP(5,2,1). In this case, Theorem 2 tells us to place an identity matrix in columns 2 and 3; the matrix $\boldsymbol{G}_1 = \boldsymbol{G}_{1,2}$ is computed in accordance with (4). We first construct $\boldsymbol{G}_{0,2}$ as follows; let

$$\boldsymbol{Z} = \begin{bmatrix}1 & 1 & 1\\ 1 & D & D^2\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & D^2 & 0\\ 0 & 0 & D^4\end{bmatrix}\begin{bmatrix}1 & 1 & 1\\ 1 & D & D^2\\ 1 & D^4 & D^8\end{bmatrix}^{-1}$$

$$= \begin{bmatrix}\dfrac{D^3}{1+D^2} & \dfrac{D+D^3}{1+D+D^2} & \dfrac{1}{1+D+D^3+D^4}\\ \dfrac{D^4}{1+D^2} & D^2 & \dfrac{1}{1+D^2}\end{bmatrix}.$$

Then

$$\boldsymbol{G}_{0,2} = \begin{bmatrix}\dfrac{D^3}{1+D^2} & \dfrac{D+D^3}{1+D+D^2} & 1 & 0 & \dfrac{1}{1+D+D^3+D^4}\\ \dfrac{D^4}{1+D^2} & D^2 & 0 & 1 & \dfrac{1}{1+D^2}\end{bmatrix},$$

and so from (4) the generator matrix for CP(5,2,1) is given by

$$\boldsymbol{G}_1 = D^{-2}\begin{bmatrix}1 & 0\\ 0 & \dfrac{1}{D}\end{bmatrix}\boldsymbol{G}_{0,2}\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & D & 0 & 0 & 0\\ 0 & 0 & D^2 & 0 & 0\\ 0 & 0 & 0 & D^3 & 0\\ 0 & 0 & 0 & 0 & D^4\end{bmatrix} = \begin{bmatrix}\dfrac{D}{1+D^2} & \dfrac{1+D^2}{1+D+D^2} & 1 & 0 & \dfrac{D^2}{1+D+D^3+D^4}\\ \dfrac{D}{1+D^2} & 1 & 0 & 1 & \dfrac{D}{1+D^2}\end{bmatrix}.$$

It is possible to shorten the construction given above by forming the generator matrix $\boldsymbol{G}_{m,r-m}$ directly, rather than first constructing $\boldsymbol{G}_{0,r-m}$. We use this "shortcut" in the following summary of the results in this section.

To construct a systematic generator matrix for CP(n, k, m), 1) compute

$$\boldsymbol{Z} = D^{-(r-m)m}\boldsymbol{\Psi}_k^{-m}\boldsymbol{Q}_{k,r}\boldsymbol{\Psi}_r^{r-m}\begin{bmatrix}\boldsymbol{Q}_{r-m,r}\\ \boldsymbol{Q}_{m,r}\boldsymbol{\Psi}_r^{n-m}\end{bmatrix}^{-1}\Gamma^m$$

where

$$\Psi_i = \mathrm{diag}[1, D, D^2, \cdots, D^{(i-1)}]$$

$$\Gamma = \mathrm{diag}[1, D, D^2, \cdots, D^{r-m-1}, D^{r-m+k}, D^{r-m+k+1}, \cdots, D^{n-1}],$$

and $\boldsymbol{Q}_{a,b}$ is given in (3); 2) let $\bar{z}_0, \bar{z}_1, \cdots, \bar{z}_{r-1}$ be the columns of $\boldsymbol{Z}$. Then set

$$\boldsymbol{G}_m = [\bar{z}_0 \bar{z}_1 \cdots \bar{z}_{r-m-1} \boldsymbol{I}_k \bar{z}_{r-m+1} \cdots \bar{z}_{r-1}].$$

B. On the Constraint Lengths of CPC Codes

A common measure of the complexity of a convolutional encoder is the constraint length. In this section we will compute the constraint length of CP(n, k, m).

Let $\boldsymbol{G}$ be a $k \times n$ matrix over $\boldsymbol{F}[D]$ and let $\{g_{ij}(D): 0 \le i \le k-1, 0 \le j \le n-1\}$ be the elements of $\boldsymbol{G}$; $g_{ij}(D)$ is a polynomial in D with binary coefficients.

Definition: The *constraint length* for the ith input of $\boldsymbol{G}$ is given by

$$\nu_i = \max_{0 \le j \le n-1} \{\deg[g_{ij}(D)]\}$$

and the *overall* constraint length of $\boldsymbol{G}$ is given by

$$\nu_G = \sum_{j=0}^{k-1} \nu_i.$$

The interpretation of ν_G is simple. Suppose that $\boldsymbol{G}$ is used as the generator of a convolutional code; that is, a message $\bar{m} \in \boldsymbol{F}^k((D))$ is mapped onto $\bar{c} = \bar{m}\boldsymbol{G}$. This generator can be immediately realized with k shift registers, the ith of which has length ν_i. Thus ν_G indicates how many memory components are necessary to implement an encoder using this obvious realization.

Definition: For a convolutional code C the *minimal constraint length* of C is the overall constraint length "simplest" polynomial generator for C; that is, if we denote the minimal constraint length of C as $N(C)$, then

$$N(C) = \min\{\nu_G: C = \{\bar{m}\boldsymbol{G}: \bar{m} \in \boldsymbol{F}^k((D))\}, \boldsymbol{G} \text{ polynomial}\}.$$

Forney [10] showed that, for any code C, no generator —either polynomial or not—can be constructed with fewer than $N(C)$ memory elements. However, any systematic generator—like the ones developed for CPC codes in Section IV-A—*can* be constructed with $N(C)$ memory elements, although this might not necessarily be possible using the obvious realization.

Fuja [19] showed that the overall constraint length for CP$(n, k, m=0)$ is $k\binom{n-k}{2}$. Abdel-Ghaffar subsequently generalized Fuja's result for all m [20]. Theorem 3 is therefore by Abdel-Ghaffar.

Theorem 3: The overall constraint length of CP(n, k, m) is given by

$$N(\mathrm{CP}(n, k, m)) = k\binom{n-k}{2} - km(n-k-1-m).$$

Proof: Forney [10] showed that the overall constraint length of an (n, k) convolutional code is equal to the maximum degree of any of the $r \times r$ minors ($r = n-k$) of a basic parity check matrix for the code. (A matrix is *basic* if it is a polynomial matrix and has a right-inverse that is also a polynomial matrix.) Furthermore, every parity check matrix has an *invariant factor decomposition*; that is, an $r \times n$ parity check matrix can be expressed as the product of an $r \times r$ basic matrix, an $r \times r$ diagonal matrix, and an $r \times n$ basic matrix that is a parity check matrix for the same code. The $r \times r$ diagonal matrix has as its diagonal elements the *invariant factors* of the original parity check matrix; if λ_i is the ith diagonal element, then $\lambda_i = \Delta_i/\Delta_{i-1}$, where Δ_i is the GCD of the $i \times i$ minors of the original matrix.

Consider the invariant factor decomposition for $\boldsymbol{H}_m$; that is, factor $\boldsymbol{H}_m$ into

$$\boldsymbol{H}_m = \Lambda \Gamma \tilde{\boldsymbol{H}}_m$$

where Λ is a basic matrix, Γ is a diagonal matrix with the invariant factors of $\boldsymbol{H}_m$ on its diagonal, and $\tilde{\boldsymbol{H}}_m$ is a basic parity check matrix for the code CP(n, k, m). Then the overall constraint length of CP(n, k, m) is the highest degree of any of the $r \times r$ minors of $\tilde{\boldsymbol{H}}_m$; since Λ is basic, $|\Lambda| = 1$ and so

$$N(\mathrm{CP}(n, k, m)) = \max\{\deg|h|: h \in Q\} - \deg(|\Gamma|),$$

where Q is the set of all $r \times r$ subarrays of $\boldsymbol{H}_m$. The determinant of Γ is the product of all the invariant factors of $\boldsymbol{H}_m$, which is just the GCD of all the $r \times r$ subarrays of $\boldsymbol{H}_m$. Therefore,

$$\begin{aligned} N(\mathrm{CP}(n, k, m)) &= \max\{\deg|h|: h \in Q\} \\ &\quad - \deg(\mathrm{GCD}\{|h|: h \in Q\}) \\ &= \max\{\deg|\boldsymbol{\Phi}_m \boldsymbol{H}_0[i_0, \cdots, i_{r-1}]\Lambda^m|: \\ &\quad 0 \le i_0 < \cdots < i_{r-1} \le n-1\} \\ &\quad - \deg[\mathrm{GCD}\{|\boldsymbol{\Phi}_m \boldsymbol{H}_0[i_0, \cdots, i_{r-1}]\Lambda^m|: \\ &\quad 0 \le i_0 < \cdots < i_{r-1} \le n-1\}] \end{aligned}$$

where $\boldsymbol{H}_0[i_0, \cdots, i_{r-1}]$, $\boldsymbol{\Phi}_m$, and Λ are as defined in Theorem 1. Since $\boldsymbol{\Phi}_m$ does not depend on the i_j's, we conclude

$$\begin{aligned} N(\mathrm{CP}(n, k, m)) &= \max\{\deg|\boldsymbol{H}_0[i_0, \cdots, i_{r-1}]\Lambda^m|: \\ &\quad 0 \le i_0 < \cdots < i_{r-1} \le n-1\} \\ &\quad - \deg[\mathrm{GCD}\{|\boldsymbol{H}_0[i_0, \cdots, i_{r-1}]\Lambda^m|: \\ &\quad 0 \le i_0 < \cdots < i_{r-1} \le n-1\}]. \end{aligned} \tag{5}$$

By inductions on r we can show $\deg|\boldsymbol{H}_0[i_0, \cdots, i_{r-1}]| = i_1 + 2i_2 + \cdots + (r-1)i_{r-1}$, and so

$$\begin{aligned} &\deg|\boldsymbol{H}_0[i_0, \cdots, i_{r-1}]\Lambda^m| \\ &\quad = (-m)i_0 + (1-m)i_1 + \cdots \\ &\qquad + (0)i_m + \cdots + (r-1-m)i_{r-1}. \end{aligned}$$

This maximized by setting $i_j = j$ for $j = 0, \cdots, m$ and $i_j = n-r+j$ for $j = m+1, \cdots, r-1$. Plugging these val-

ues into our expression for $\deg|\boldsymbol{H}[i_0,\cdots,i_{r-1}]\Lambda^m|$, we find

$$\begin{aligned}\max\{\deg|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]\Lambda^m|: 0\le i_0<\cdots<i_{r-1}\le n-1\}\\ =[n(r-m-1)(r-m)12]\\ -[(r-m)(r-m-1)(r-m+1)/6]\\ -[m(m-1)(m+1)/6].\end{aligned}\tag{6}$$

We now find the degree of the GCD of all the $r\times r$ minors of $\boldsymbol{H}_0\Lambda^m$. Any minor of $\boldsymbol{H}_0\Lambda^m$ is a minor of $\boldsymbol{H}_0$ times a power of D. We can express the GCD of the minors of $\boldsymbol{H}_0$ as $D^a p(D)$ for some integer a and some polynomial $p(D)$ such that $p(0)=1$. Therefore, the GCD of the minors of $\boldsymbol{H}_0\Lambda^m$ is $D^b p(D)$ for some integer b; since the delay of $p(D)$ is zero, b is just the delay of the GCD of the minors of $\boldsymbol{H}_0\Lambda^m$. (*Note*: The *delay* of a Laurent series $f(d)$, denoted $\mathrm{del}[f(D)]$, is the lowest power of D in $f(D)$.) Therefore,

$$\begin{aligned}\deg\left[\mathrm{GCD}\left\{|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]\Lambda^m|: 0\le i_0<\cdots\le i_{r-1}\right\}\right]\\ =\mathrm{del}\left[\mathrm{GCD}\left\{|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]\Lambda^m|:\right.\right.\\ \left.\left. 0\le i_0<\cdots\le i_{r-1}\right\}\right]+\deg[p(D)].\end{aligned}\tag{7}$$

The delay of the GCD of the minors of $\boldsymbol{H}_0\Lambda^m$ is the smallest delay taken over all such minors. Induction on r shows that

$$\mathrm{del}|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]|=(r-1)i_0+(r-2)i_1+\cdots+i_{r-2},$$

and so

$$\begin{aligned}\mathrm{del}|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]\Lambda^m|\\ =(r-m-1)i_0+(r-m-2)i_1\\ +\cdots+(0)\mathrm{i}_{r-m-1}+\cdots+(1-m)i_{r-2}+(-m)i_{r-1}\end{aligned}$$

The minimum such delay is clearly obtained by setting $i_j=j$ for $j=0,\cdots,r-m-1$ and $i_j=n-r+j$ for $j=r-m,\cdots,r-1$. Plugging these values into our expression for $\mathrm{del}|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]\Lambda^m|$, we find

$$\begin{aligned}\mathrm{del}\left[\mathrm{GCD}\left\{|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]\Lambda^m|: 0\le i_0<\cdots\le i_{r-1}\right\}\right]\\ =[(r-m)(r-m-1)(r-m-2)/6]\\ -[m(m+1)(3n-m-2)/6].\end{aligned}$$

Finally, we must find $\deg[p(D)]$. By definition, $\deg[p(D)]$ is just the degree of the GCD of the minors of $\boldsymbol{H}_0$ minus the delay of the GCD of the minors of $\boldsymbol{H}_0$. Using Fuja's result for $m=0$, we can show that the GCD of the minors of $\boldsymbol{H}_0$ is just $|\boldsymbol{H}_0[0,1,\cdots,r-1]|$, and so

$$\begin{aligned}\deg[p(D)]=\deg|\boldsymbol{H}_0[0,\cdots,r-1]|-\mathrm{del}|\boldsymbol{H}_0[0,\cdots,r-1]|\\ =[(r-2)(r-1)(2r-3)/3]\\ -\left[(r-1)^2(r-4)/2\right].\end{aligned}$$

Using (7) we have now shown

$$\begin{aligned}\deg\left[\mathrm{GCD}\left\{|\boldsymbol{H}_0[i_0,\cdots,i_{r-1}]\Lambda^m|: 0\le i_0<\cdots\le i_{r-1}\right\}\right]\\ =[(r-m)(r-m-1)(r-m-2)/6]\\ -[m(m+1)(3n-m-2)/6]\\ +[(r-2)(r-1)(2r-3)/3]-\left[(r-1)^2(r-4)/2\right].\end{aligned}\tag{8}$$

Substituting (6) and (8) into (5) and simplifying we obtain the desired result. Q.E.D.

V. Decoding of Cross Parity Check Codes

In this section we will demonstrate how the geometric geometric regularity inherent in CPC codes can be used for decoding. Specifically, we will present a general erasure decoding algorithm for CP(n,k,m) and a double-error correcting algorithm for CP$(n,n-4,1)$. Note that Piret and Krol [13] gave error-correcting algorithms that can be used for *all* MDS convolutional codes; our contributions here are significant in that they show how the geometric regularity in CPC codes can be used to fashion simple decoding procedures.

The algorithms in this section are examples of syndrome decoding. In the case of CP(n,k,m), the syndrome of a received n-tuple are determined by the parities of the bits falling on lines of slope $1/x$, where $m\le x\le m-(n-k-1)$. Thus let $S^{(x)}(i,j)$ be defined as the modulo-two sum of the bits lying on the line of slope $1/x$ passing through the jth bit on track i. That is, if

$$A_n=\left\{a_{ij}: 0\le i\le n-1,-\infty<j<+\infty, a_{ij}\in\{0,1\}\right\}$$

are the tape contents received at the decoder, then

$$S^{(x)}(i,j)=\sum_{l=0}^{n-1}a_{l,\,j-(i-l)x}.\tag{9}$$

We will show in Section V-A how these syndromes can be used to correct $n-k$ erasures. In Section II-B, the syndromes of CP$(n,n-4,2)$ will be used to correct two errors.

The underlying concept of these algorithms is simple. In each case we will construct a piecewise linear partition running across the tape, passing through exactly one bit on each track; we will show how this line, called a *decoding boundary*, can be constructed to have the following property. If everything on one side of the line is assumed to be correct—that is, exactly as it was written by the encoder—then the syndromes can be used to discern what was written into the locations lying on the boundary. As mentioned in Section II, one always assumes that zeros are written at the beginning of a tape, and so we always have a "starting place" where the tape contents are known. We use this information to make corrections to bits lying on a boundary and then advance the boundary by one bit. Continuing in this vein, one can correct all tracks, one boundary at a time.

A. Erasure Decoding of CP(n, k, m)

In this section we will show how CP(n,k,m) can be used to correct $n-k$ erased tracks. As noted earlier, we will use decoding boundaries to achieve our goal; furthermore, since we are concerned only with decoding what is on the erased tracks—the unerased tracks being presumed correct—the decoding boundary need only be defined on those erased tracks. It is easiest to see how this is done by

first examining a particular example, and then generalizing the technique.

Consider the code CP$(n, n-5, 2)$. This code is defined by five slope constraints; the bits falling on any line of slope $1/x$, $-2 \le x \le 2$, must sum to zero modulo two. In Fig. 3 it is shown how these constraints can be used to create a decoding boundary. Assume that tracks $e_0 < e_1 < e_2 < e_3 < e_4$ are erased and that all the data to the left of the marked bits on the erased tracks are correct. To decode the marked bits, we first use the $-1/2$ and $1/2$ constraints to correct the emboldened bits on tracks e_0 and e_4, respectively. Having corrected those bits, we then use the -1 and $+1$ constraints to correct the erased bits on tracks e_1 and e_3. Finally, we can then decode the bit lying on e_2 by computing the vertical parity. If one assumes that there were no errors on any of the unerased tracks, then the decoder has correctly estimated all of the emboldened bits, and the boundary can be moved one bit to the right.

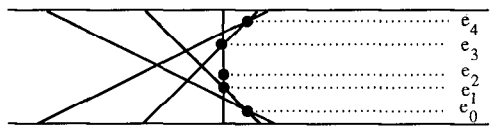

Fig. 3. Construction of erasure decoding boundary for CP$(n, n-5, 2)$.

This technique easily generalizes. One works inward from the edges of the decoding boundary, using the bit just decoded to get the next one on the boundary; then the vertical parity check determines the final bit.

The general decoding algorithm is given in flowchart form in Fig. 4. In this description we assume that the tape has been encoded with CP(n, k, m), and a possibly corrupted version is read back. The notation used is the one given above; a_{ij} is the value of the jth bit on track i, and $S^{(x)}(i, j)$ is the modulo-two sum of the bits falling on the line of slope $1/x$ passing through that bit. Furthermore, we assume that tracks, e_i for $i = 0, 1, \cdots, r-1 (r = n-k)$ are erased, and $e_i < e_j$ for $i < j$. Thus the goal is to determine the value of $a_{e_i, j}$ for $0 \le i \le r-1$ and $j \ge 0$. (As before, we assume that $a_{ij} = 0$ for $j < 0$.)

B. Error Decoding of CP$(n, n-4, 1)$

In this section we will show how the four parity tracks of CP$(n, n-4, 1)$ can be used to correct any number of bit errors, provided they all occur on only two tracks. Assume that a tape is encoded with CP$(n, n-4, 1)$ and that a possibly corrupted version is read back. As before, let a_{ij} denote the contents of the jth bit on the ith track, and let $S^{(x)}(i, j)$ be the modulo-two sum of the bits lying on a line of slope $1/x$ passing through that bit, as in (9). The decoding boundary for this algorithm is a vertical line; that is, at the lth step of the algorithm we assume that $\{a_{ij}: 0 \le i \le n-1, j < l\}$ are correct—as written by the encoder— and the algorithm will decode a_{il} for $i = 0, 1, \cdots, n-1$.

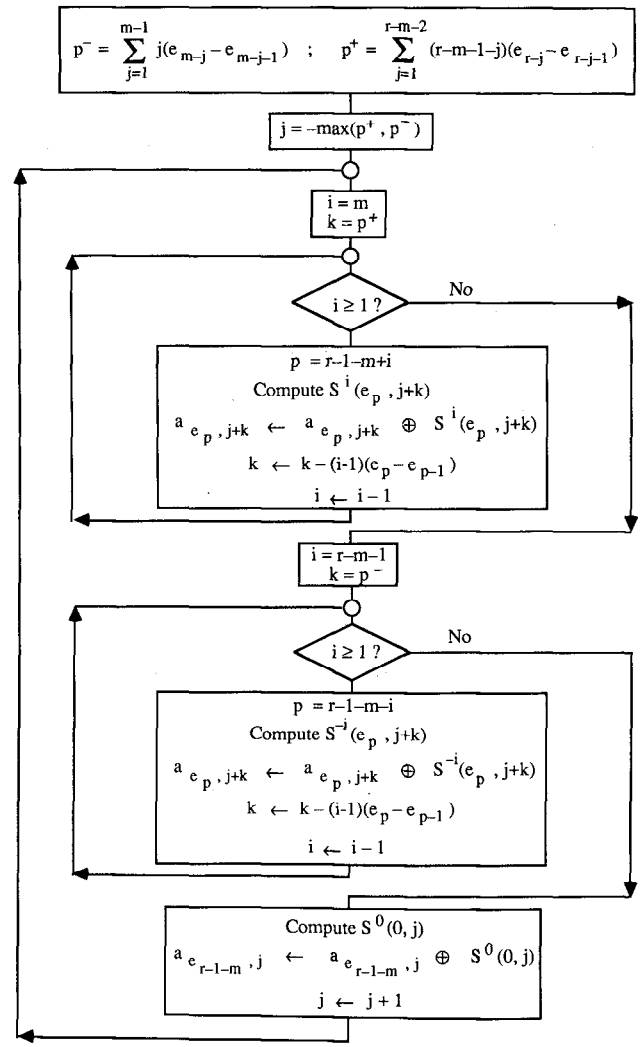

Fig. 4. Erasure decoding algorithm for CP(n, k, m).

To this end, define for all $j \ge 0$ and for $x \in \{-2, -1, 1\}$ the set

$$\boldsymbol{P}_j^{(x)} = \{i: 0 \le i \le n-1, S^{(x)}(i, j) = 1\}.$$

Simply put, $\boldsymbol{P}_j^{(x)}$ contains the numbers of those tracks that are "flagged" by a slope $1/x$ parity check at time j. Next, define the following pointers:

$$P_j^{(0)} = \begin{cases} \max \boldsymbol{P}_j^{(1)}, & \text{if } |\boldsymbol{P}_j^{(1)}| > 0 \\ -1. & \text{otherwise} \end{cases}$$

$$P_j^{(-1)} = \begin{cases} \min \boldsymbol{P}_j^{(-1)}, & \text{if } |\boldsymbol{P}_j^{(-1)}| > 0 \\ -1 & \text{otherwise} \end{cases}$$

$$P_j^{(-2)} = \begin{cases} \min \boldsymbol{P}_j^{(-2)}, & \text{if } |\boldsymbol{P}_j^{(-2)}| > 0 \\ -1, & \text{otherwise.} \end{cases}$$

The significance of these pointers is best seen by example; in Fig. 5 their values are shown for a variety of error configurations on a ten-track magnetic tape. (A bit error is denoted by a shaded circle.) Heuristically, the pointers

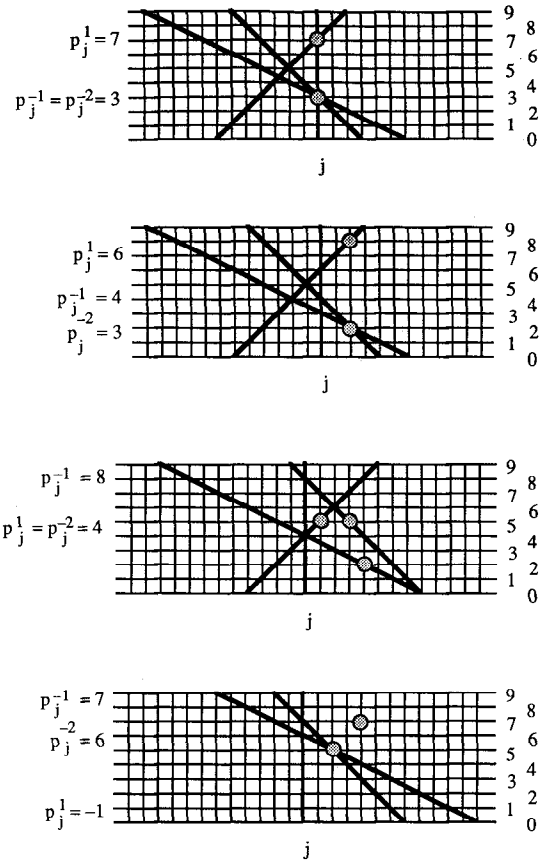

Fig. 5. Pointer values for variety of error nonfigurations.

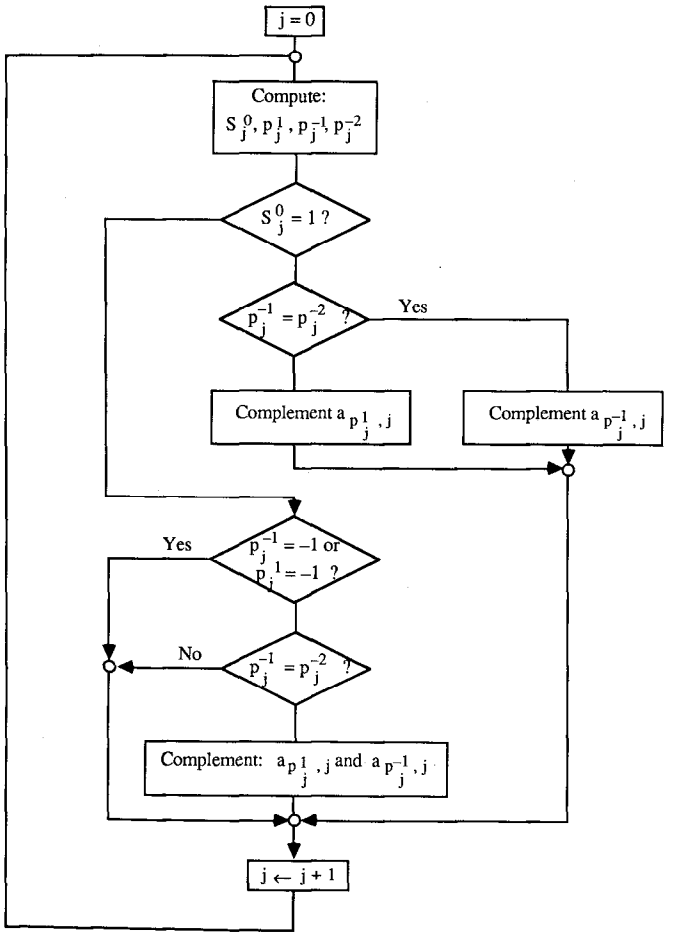

Fig. 6. Double error correcting algorithm for CP$(n, n-4, 1)$.

indicate which track is the first one "flagged" as the three skewed parity check lines slide over the decoding boundary; if no tracks are flagged, then the pointers are assigned the arbitrary value of -1. Note that if errors line up in a certain way, they can be "shielded" from the pointers. In the third example in Fig. 5, two bit errors fall on a line of slope -1 and so are hidden from $S^{(-1)}(i, j)$; similarly, in the fourth example the set $\boldsymbol{P}_j^{(1)}$ is empty because the two errors that occur fall on a line of slope one.

Fig. 6 gives an algorithm that makes use of these pointers to describe a two-error-correcting decoder for CP$(n, n-4, 1)$. At each iteration, the decoder first checks the value of $S_j^{(0)} \equiv S^{(0)}(i, j)$. (We use the shorthand notation $S_j^{(0)}$ because $S^{(0)}(i, j)$ obviously does not depend on i.) If $S_j^{(0)} = 1$, then by our assumption that at most two tracks contain errors, we know that there is exactly one bit error in the jth column of the tape. Furthermore, either $p_j^{(1)}$ or $p_j^{(-1)}$ (or possibly both) will point to the corrupted bit; it is easy to verify that $p_j^{(-1)}$ points to the error if and only if $p_j^{(-1)} = p_j^{(-2)}$.

If $S_j^{(0)} = 0$, then there are either no errors in the jth column or there are two errors. If there are two errors, then both $\boldsymbol{P}_j^{(1)}$ and $\boldsymbol{P}_j^{(-1)}$ will be nonempty *and* both $p_j^{(-1)}$ and $p_j^{(-2)}$ will point to the same track. If both of these conditions hold, then the two errors will be indicated by $p_j^{(1)}$ and $p_j^{(-1)}$; if either condition does not hold, then there are no errors in the jth column.

VI. Shortened Cross Parity Check Codes

In this section we will construct new MDS convolutional codes from cross parity check codes. Our motivation is to reduce the length of the parity check lines defining our geometric codes. There are two reasons for this. First, there is an inherent decoding delay incurred by long parity check lines because the tape drive's read head must pass over the entire length of the line before it can compute the associated syndrome. A second problem with very long parity check lines is that they increase the length of the error-free "window" that must occur between error bursts. For instance, the erasure decoding algorithm given in Section V-A is capable of correcting up to $n-k$ erased tracks; however, an error-free interval is required to change the erasure locations, and the longer the parity check lines, the longer the window required.

We wish to construct parity check matrices that reflect short geometric constraints; this suggests keeping the highest power of D in the parity check matrix relatively small. To do this, we will use the technique of parity check matrix reduction, first introduced by Piret and Krol [13].

Definition: For an arbitrary matrix A over $\boldsymbol{F}[D]$ and for any $\pi(D) \in \boldsymbol{F}[D]$, define another matrix $A^{[\pi(D)]}$ whose entries are the remainder of the division of the element, in

A by $\pi(D)$;

$$A = \left[a_{ij}(D)\right] \rightarrow A^{[\pi(D)]} = \left[\text{rem}\left(\frac{a_{ij}(D)}{\pi(D)}\right)\right].$$

Definition: For any convolution code C defined in terms of a parity check matrix $\boldsymbol{H}$, and for any $\pi(D) \in \boldsymbol{F}[D]$, define C mod $\pi(D)$ as the code described by the parity check matrix $\boldsymbol{H}^{[\pi(D)]}$.

In this section we are interested in finding polynomials $\pi(D)$ such that CP(n, k, m) modulo $\pi(D)$ is still MDS.

Theorem 4: CP(n, k, m) mod $\pi(D)$ is MDS for any irreducible, primitive polynomial $\pi(D) \in \boldsymbol{F}[D]$ provided $\deg[\pi(D)] \geq \log_2(n+1)$.

Proof: The proof of the theorem is based on [13, theorem 1]; it consists of considering D not as an indeterminate, but instead as a primitive root of $\pi(x)$. Then each element of $\boldsymbol{H}_m$ is an element of $\text{GF}(2^{\deg[\pi(x)]}) = \boldsymbol{F}[x]/\pi(x)$, and it can be shown that this matrix is totally nonsingular. (In fact, it is a parity check matrix for a Reed–Solomon code.) Of course the matrix $\boldsymbol{H}_m^{[\pi(D)]}$ is equal to $\boldsymbol{H}_m$ over this field, meaning it, too, is totally nonsingular. Considering D as an indeterminate again, this means that each $r \times r$ minor of $\boldsymbol{H}_m^{[\pi(D)]}$ is nonzero modulo $\pi(D)$ and so nonzero. Q.E.D.

Theorem 4 lets us markedly decrease the delay of any CPC code. Unfortunately, the technique used to achieve this completely destroys the nice geometric properties described in Section II; shortening $\boldsymbol{H}_m$ with some arbitrary primitive irreducible polynomial $\pi(D)$ "folds" the parity check lines in a very irregular way.

The desire to retain some semblance of the geometric properties we used to define the CPC codes leads us to consider the codes CP(n, k, m) mod (D^j+1); heuristically, the parity check lines for CP(n, k, m) mod (D^j+1) are formed by "breaking" the parity checks for CP(n, k, m) and "sliding" them backwards. An example of this procedure is given in Fig. 7.

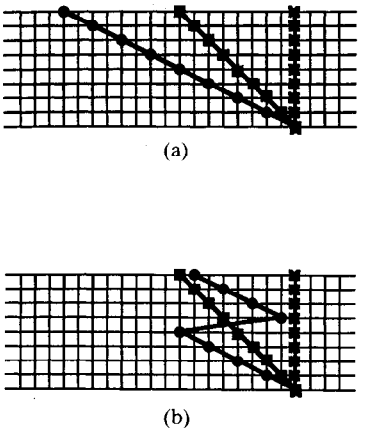

Fig. 7. Example of effect of reducing CPC code by D^j+1. (a) Parity check lines for CP(9,6,0). (b) Parity check lines for CP(9,6,0) mod D^9+1.

Theorem 5: CP(n, k, m) modulo D^j+1 is MDS for all odd $j \geq n$.

Proof: We must show that every $r \times r$ subarray of $\boldsymbol{H}_m^{[\pi(D)]}$ is nonzero. It is sufficient, as in Theorem 4, to show that every $r \times r$ subarray of $\boldsymbol{H}_m$ is nonzero modulo D^j+1. However, from the proof of Theorem 1 we know that every $r \times r$ minor of $\boldsymbol{H}_m$ can be written as the corresponding minor of $\boldsymbol{H}_0$ times a power of D. Thus each $r \times r$ minor of $\boldsymbol{H}_m$ is nonzero modulo D^j+1 if and only if each $r \times r$ minor of $\boldsymbol{H}_0$ is nonzero modulo D^j+1. However from Theorem 1, the minor obtained by selecting columns $i_0, \cdots, i_{r-1}$ is of the form

$$\det\left(\boldsymbol{H}_0[i_0, \cdots, i_{n-k-1}]\right) = \prod_{a>b}\left(D^{i_a} + D^{i_b}\right).$$

Since j is odd, D^j+1 has a primitive jth root of the unity that is not a root of $D^{i_a} + D^{i_b}$ for all a and b. Therefore, the determinant cannot be nonzero modulo D^j+1 and the theorem is proved. Q.E.D.

VII. Future Work

The following unresolved issues concerning cross parity check codes warrant further consideration.

- While a general erasure-correcting algorithm for CP(n, k, m) was presented in Section V the "companion piece"—a general error correcting algorithm—is yet to be found. From the results presented in Section III, we know that CP(n, k, m) is capable of correcting any number of errors provided they occur on at most $\lfloor (n-k)/2 \rfloor$ different tracks. At this point, we have developed algorithms capable of achieving this only for $n - k \leq 4$.
- Decoding algorithms that take advantage of the geometric regularity of the "shortened" codes of Section VI have yet to be developed.

Acknowledgment

The authors would like to thank Khaled Abdel-Ghaffar for his permission to publish the proof of Theorem 3. We would also like to thank Reviewer "C" suggestions regarding the simplification of Section VI.

Appendix

Theorem 3: The matrix $\boldsymbol{G}_{m,x}$ as defined in Section IV-A is realizable for $x = r - m$.

Proof: First consider the case $m = 0$. From the definition of the generator and parity check matrices, it is easy to verify that $\boldsymbol{G}_{0,x}\boldsymbol{H}_0^T = 0$. If $\boldsymbol{Z}$ is defined as the $k \times r$ matrix consisting of $\boldsymbol{G}_{m,x}$ with the identity matrix removed, then the above equation can be reformulated as

$$\boldsymbol{G}_{0,x}\boldsymbol{H}_0^T = \boldsymbol{Z}\boldsymbol{P} + \boldsymbol{Q}_{k,r}\boldsymbol{\Psi}_r^x = 0$$

where $\boldsymbol{\Psi}, \boldsymbol{Q}_{a,b}$ and $\boldsymbol{P}$ are as given in Section IV-A. From this we arrive at (2) from Section IV-A:

$$\boldsymbol{Z} = \boldsymbol{Q}_{k,r}\boldsymbol{\Psi}_r^x\boldsymbol{P}^{-1}. \quad \text{(A1)}$$

The goal, then, is to pick x so that every element of $\boldsymbol{Z}$ as defined in (A1) has, after reduction to lowest terms, a denominator with a nonzero constant term.

Consider two matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$ over some field; let the dimensions of $\boldsymbol{A}$ be $k \times r$, and let those of $\boldsymbol{B}$ be $r \times r$. The elements of the two matrices are given by

$$\boldsymbol{A} = \begin{bmatrix} 1 & \alpha_0 & \alpha_0^2 & \alpha_0^3 & \cdots & \alpha_0^{r-1} \\ 1 & \alpha_1 & \alpha_1^2 & \alpha_1^3 & \cdots & \alpha_1^{r-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \alpha_{k-1} & \alpha_{k-1}^2 & \alpha_{k-1}^3 & \cdots & a_{k-1}^{r-1} \end{bmatrix}$$

and

$$\boldsymbol{B} = \begin{bmatrix} 1 & \beta_0 & \beta_0^2 & \beta_0^3 & \cdots & \beta_0^{r-1} \\ 1 & \beta_1 & \beta_1^2 & \beta_1^3 & \cdots & \beta_1^{r-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \beta_{r-1} & \beta_{r-1}^2 & \beta_{r-1}^3 & \cdots & \beta_{r-1}^{r-1} \end{bmatrix}.$$

We wish to find a "nice" form for the matrix product $\boldsymbol{AB}^{-1}$. To this end, let $\{\gamma_{ij}\}$ be the elements of B^{-1}, and define the r polynomials

$$\gamma_j(y) = \sum_{i=0}^{r-1} \gamma_{ij} Y^i$$

for $j = 0, 1, \cdots, r-1$. Now from the fact that $\boldsymbol{BB}^{-1} = \boldsymbol{I}$, we conclude that $\gamma_j(\beta_i) = \delta_{ij}$, the Kronecker delta. This however uniquely determines $\gamma_j(y)$ to be

$$\gamma_j(y) = \prod_{\substack{i=0 \\ i \neq j}}^{r-1} \frac{y - \beta_i}{\beta_j - \beta_i}.$$

Taking the product $\boldsymbol{AB}^{-1}$ just involves evaluating these polynomials at the different values of α_i. Specifically, if we let $\{\pi_{ij}: 0 \le i \le k-1, 0 \le j \le r-1\}$ be the elements of $\boldsymbol{AB}^{-1}$, then

$$\pi_{ij} = \prod_{\substack{l=0 \\ l \neq j}}^{r-1} \frac{\alpha_i - \beta_l}{\beta_j - \beta_l}.$$

To see how all this relates to (A1), we note that

$$\boldsymbol{Q}_{k,r}\boldsymbol{\Psi}_r^x = \begin{bmatrix} 1 & D^x & D^{2x} & \cdots & D^{(r-1)x} \\ 1 & D^{x+1} & D^{2(x+1)} & \cdots & D^{(r-1)(x+1)} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & D^{x+k-1} & D^{2(x+k-1)} & \cdot & D^{(r-1)(x+k-1)} \end{bmatrix},$$

and so $\boldsymbol{Q}_{k,r}\boldsymbol{\Psi}_r^x\boldsymbol{P}^{-1}$ is a product of the form described above. Therefore, if $\{z_{ij}: 0 \le i \le k-1, 0 \le j \le r-1\}$ are the elements of $\boldsymbol{Z}$, (A1) can be reformulated as

$$z_{ij} = \begin{cases} \displaystyle\prod_{\substack{a=0 \\ a \neq j}}^{x-1} \frac{D^{x+i} + D^a}{D^j + D^a} \prod_{b=x}^{r-1} \frac{D^{x+i} + D^b}{D^j + D^b}, & \text{if } j < x \quad \text{(A2a)} \\ \displaystyle\prod_{a=0}^{x-1} \frac{D^{x+i} + D^a}{D^{j+k} + D^a} \prod_{\substack{b=x \\ b \neq j}}^{r-1} \frac{D^{x+i} + D^{b+k}}{D^{j+k} + D^{b+k}}, & \text{if } j \ge x. \quad \text{(A2b)} \end{cases}$$

We recall that the goal is to find an x such that all the z_{ij} are realizable. This suggest factoring (A2):

$$z_{ij} = \begin{cases} D^{(x/2)-(j/2)+(r-x)(i+x)-j(r-j)} \displaystyle\prod_{a=1}^{j} \frac{1+D^{x+i-a+1}}{1+D^a} \\ \quad \times \displaystyle\prod_{b=1}^{x-1-j} \frac{1+D^{x-b+i+j}}{1+D^b} \prod_{c=0}^{r-1-x} \frac{1+D^{k-i+c}}{1+D^{c+k+x-j}}, & \text{if } j < x \\ D^{(x/2)-(j/2)+(r-1-x)(i+x-k)-j(r-1-j)} \displaystyle\prod_{a=0}^{x-1} \frac{1+D^{x+i-a}}{1+D^{j+k-a}} \\ \quad \times \displaystyle\prod_{b=0}^{j-1-x} \frac{1+D^{b+k-i}}{1+D^{j-x-b}} \prod_{c=1}^{r-1-j} \frac{1+D^{k-i+j-x+c}}{1+D^c}, & \text{if } j \ge x. \end{cases}$$

(Note: We take the convention that $\binom{a}{2} = 0$ for $a = 0$ or $a = 1$, and that $\prod_{i=y}^{x} a_i = 1$ for $y > x$.)

The multiple products above contain a nonzero constant term in the denominator even after reduction to lowest terms. Thus z_{ij} is realizable if and only if $e_{ij} \ge 0$, where

$$e_{ij} = \begin{cases} \binom{x}{2} - \binom{j}{2} + (r-x)(i+x) - j(r-j), & \text{if } j < x \\ \binom{x}{2} - \binom{j}{2} + (r-1-x)(i+x-k) - j(r-1-j), & \text{if } j \ge x. \end{cases}$$

In fact, it is easily seen through elementary calculus that if we set $x = r$ (and thus $j < x$ for all j), then e_{ij} is nonnegative for all i and j. Thus $\boldsymbol{G}_{0,r}$ is realizable, and so it is a valid systematic generator for CP($n, k, m = 0$).

A similar process leads to construction of a systematic generator for CP(n, k, m) for all m, $0 \le m \le m - k - 1$. As shown in Section IV-A, $\boldsymbol{G}_{m,x}$ can be expressed as

$$\boldsymbol{G}_{m,x} = D^{-xm}\boldsymbol{\Psi}_k^{-m}\boldsymbol{G}_{0,x}\boldsymbol{\Psi}_n^m.$$

Using this formulation and proceeding as before, if we define the $k \times r$ matrix $\boldsymbol{Z}$ to consist of $\boldsymbol{G}_{m,x}$ with the identity matrix removed, and let $\{z_{ij}: 0 \le i \le k-1, 0 \le j \le r-1\}$ be the elements of that matrix, then

$$z_{ij} = D^{f_{ij}} \frac{a_{ij}(D)}{b_{ij}(D)}$$

where $b_{i,j}(0) = 1$ and

$$f_{i,j} = \begin{cases} \binom{x}{2} - \binom{j}{2} + (r-x)(i+x) - j(r-j) - (x-j+i)m, & \text{if } j < x \\ \binom{x}{2} - \binom{j}{2} + (r-1-x)(i+x-k) - j(r-1-j) - (x-j-k+i)m, & \text{if } j \geq x. \end{cases}$$

Simple calculations reveal that if $x = r - m$, then $f_{i,j} \geq 0$ for all i and j, and so $\boldsymbol{G}_{m,r-m}$ is realizable. This proves Theorem 2. Q.E.D.

References

[1] A. M. Patel and S. J. Hong, "Optimal rectangular code for high density magnetic tapes," *IBM J. Res. Develop.*, vol. 18, no. 6, pp. 579–588, Nov. 1974.

[2] E. R. Berlekamp, "Algebraic codes for improving the reliability of tape storage," presented at the 1975 Nat. Computer Conf.

[3] A. M. Patel, "Error recovery scheme for the IBM 3580 mass storage system," *IBM J. Res. Develop.*, vol. 24, no. 1, pp. 32–42, Jan. 1980.

[4] P. Prusinkiewicz and S. Budlowski, "A double track error-correction code for magnetic tape," *IEEE Trans. Comput.*, vol. C-19, pp. 642–645, June 1976.

[5] M. Patel, "Multitrack error correction with cross-parity-check coding," *IEEE Int. Symp. Information Theory*, Brighton, England, June 23–28, 1985.

[6] ____, "Adaptive cross-parity code for a high-density magnetic tape subsystem," *IBM J. Res. Develop.*, vol. 29, no. 6, pp. 546–562, Nov. 1985.

[7] M. Blaum, "A family of error-correcting codes for magnetic tapes," Res. Rep., IBM Almaden Research Center, San Jose, CA.

[8] E. Scott and D. Goetschel, "One check bit per word can correct multibit errors," *Electronics*, pp. 130–134, May 5, 1981.

[9] A. Shiozaki, "Proposal of a new coding pattern in orchard scheme," *Inform. Contr.*, vol. 51, pp. 209–215, 1981.

[10] G. D. Forney, "Convolutional Codes I: Algebraic structures," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 720–738, Nov. 1970.

[11] I. N. Herstein, *Topics in Algebra*, 2nd ed., New York: Wiley, 1975.

[12] E. D. Nering, *Linear Algebra and Matrix Theory*, 2nd ed., New York: Wiley, 1970.

[13] P. Piret and T. Krol, "MDS convolutional codes," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 224–232, Mar. 1983.

[14] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error Correcting Codes*. Amsterdam, The Netherlands: North Holland, 1977.

[15] W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes*. Cambridge, MA: MIT Press, 1972.

[16] V. Pless, *Introduction to the Theory of Error-Correcting Codes*. New York: Wiley, 1982.

[17] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[18] A. G. Kurosh, *Lectures on General Algebra*. New York: Chelsea, 1963.

[19] T. E. Fuja, "On the structure and decoding of cross parity check codes for magnetic tape," Ph.D. dissertation, Cornell Univ., May 1987.

[20] K. Abdel-Ghaffar, personal correspondence.